Keysight Technologies

Maximizing Test Coverage with *Medalist* VTEP v2.0 – VTEP, iVTEP, and NPM

White Paper

KEYSIGHT
TECHNOLOGIES

Introduction

This paper describes how to get the most from the Keysight Technologies, Inc. industry-leading vectorless test innovation, the *Medalist* VTEP v2.0 vectorless test suite.

VTEP v2.0 includes the original VTEP as its core and has its performance extended through iVTEP and Network Parameter Measurement (NPM). In short, the entire VTEP v2.0 test suite includes:

- VTEP
- iVTEP
- Network Parameter Measurement

VTEP is a combination of new hardware and software techniques, which enable improvements over the noise characteristics, stability, accuracy, and repeatability of TestJet measurements. This allows Keysight board test systems using VTEP to make reliable readings at much lower values than the original TestJet. This makes it possible to test pins on devices and connectors that TestJet previously could not.

Taking this innovation further, iVTEP extends the measurement capability of VTEP. It is enabled automatically by the 05.41 software upgrade. With iVTEP, more accurate measurements can be made on integrated circuits (ICs) using existing VTEP hardware. These measurements are less dependent on lead-frame geometries than VTEP or TestJet. AutoDebug has been updated to add iVTEP capability automatically on a per-pin basis. Test development, hardware, and fixturing are the same as those for VTEP.

VTEP innovations continue with NPM (Network Parameter Measurement) technology. It is enabled automatically by the 07.00 software upgrade. With NPM, opens on connector and socket power and ground pins are covered. This is the first time this capability is available on any ICT platform. Opens detection on connector and socket power and ground pins will become critical as speeds of PCAs increase. NPM capability has been added to AutoDebug. Hardware and fixturing are the same as VTEP. NPM coverage can be added to connectors in existing fixtures and programs or those developed from scratch with a single additional test development step.

As of the 07.10 software upgrade, two additional NPM models have been added for the LGA 775 and LGA 1207 CPU sockets, and throughput has been improved by up to 75%.

By using VTEP v2.0, the most immediate benefit which the end-user can get is coverage for more pins even as they get packed into ever shrinking integrated circuits with higher speed connector and sockets.

Test Techniques

The VTEP technique is a technological enhancement over TestJet. There are several factors which influence the values of capacitance measured during the test. One of these is repeatability of the measurements where noise is present.

VTEP addresses this problem with advanced measurement hardware and improved software algorithms to dramatically improve the measurements.

As a result, the noise signature of VTEP (standard deviation of the measurements) is 3 to 7 times better than TestJet, which was already better than other vectorless test solutions in the industry.

Because of this improved stability, measurements down to single-digit femtofarads can now be made. As the geometries of device packages and connectors have shrunk, the resulting capacitance of the pins to the sensor plate has also shrunk. VTEP can measure these smaller capacitance values reliably.

iVTEP is a new breakthrough. For the first time, the measured capacitance value is less dependent on shrinking lead-frame geometries. iVTEP will work only on devices with little or no lead-frames (like uBGAs and flip-chips) and devices under heat spreaders and ungrounded heatsinks. It will not work on connectors. Its noise signature is up to 7 times better than VTEP, which is in turn 3 to 7 times better than TestJet. iVTEP is a Keysight proprietary and patent-pending technique.

NOTE: There is a clear distinction between an open and good joints using VTEP as the measurement allows values to be effectively measured down to 5 femfofarads (fF). In addition, more accurate and stable measurements can be achieved when iVTEP is used as the open joint is measured close to 0 fF.

NPM is another new breakthrough. For the first time, connector and socket power and ground opens can be detected. NPM will work only on connectors and sockets. It relies on making network parameter measurements of the target connectors (series resistances, series and coupled inductances, coupled capacitances) to determine opens on power and ground pins. This coverage will become increasingly important as PCA data rates increase. These network parameters are described in connector models that are supplied in an updated library directory as part of the 7.0 software release.

Throughput

Making reliable single digit capacitance measurements takes time. Not all pins require this level of stability. VTEP supports both an enhanced mode (slower, best performance) and a fast mode.

The fast mode is as fast as TestJet in fast mode (throughput adjustment 1) but with measurement stability equal to TestJet in slow mode (throughput adjustment 0). To maximize throughput, VTEP automatically evaluates the measurements as it runs. It makes as many measurements as possible using the faster mode but will shift to enhanced mode when necessary.

This decision is based on both the magnitude of the reading and its proximity to the limits. This means that the limits can affect the throughput of the test. Therefore, it is best to let the AutoDebugger set limits since it knows how to maximize throughput.

The rules for determining which mode to use are complex. If you have a compelling need to set thresholds manually, try to set them more than 15 femtofarads from the mean of the measurements. Then the fast/enhanced mode cross-over point will be roughly 15-20 fF.

iVTEP measurements take more than twice as long as enhanced mode VTEP pins. iVTEP capability is set on a per-pin, per-device or per-test basis by the “throughput adjustment 2” flag.

This flag is added automatically by the AutoDebugger. Since this flag is the slowest (but most accurate) of all options, it is best to let the AutoDebugger determine which pins require it.

NPM measurements for power and ground pins for targeted connectors are made at 240 pins per second. When estimating throughput at 7.0, test time is approximately 10ms per device, plus:

- Fast mode pins (>20fF): 500 pins per second
- Enhanced mode pins (<20fF and NPM power and ground pins): 240 pins per second
- iVTEP pins (throughput adjustment2): 100 pins per second

With software version 07.10, throughput has been improved by up to 75%.

Fixturing Considerations for VTEP

Amplifier selection

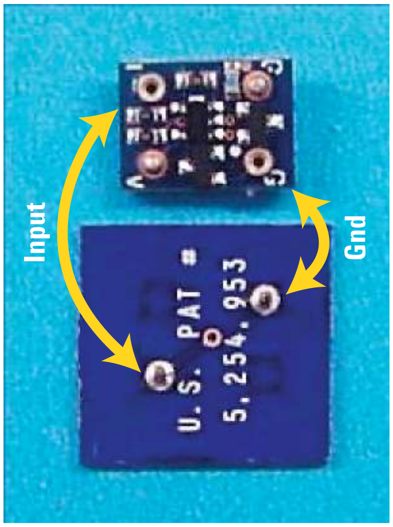
As with TestJet, there are two sizes of the amplifier board that connect to the sensor plates. It is important to note that only the standard-size VTEP amplifiers will demonstrate measurement improvements. The smaller amps (sometimes called PolarityCheck amps) do not have room for the additional circuitry. Therefore, these smaller amps have the same performance as the original TestJet. When building a fixture, try to use the standard size amplifier if possible. See the section below on customizing sensor plates.

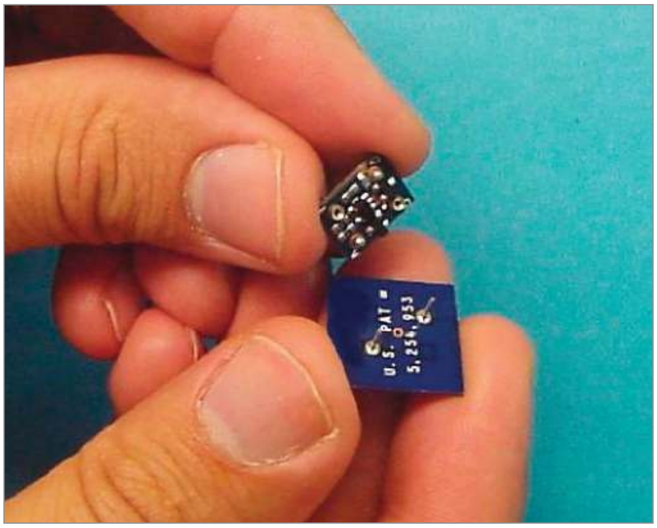
Mux card limits

The VTEP mux cards (both versions) draw more current than the original TestJet mux cards. Up to 15 of the older mux cards could be connected to a module, but with VTEP the limit is now 6 cards connected to a single module. Additional mux cards can be connected to other modules if more are needed. This gives a maximum number of devices that can be tested as 4Modules*6MuxCards*64 DevicesPerMux = 1536 devices.

Assembly

Connecting the sensor plates to the amp boards is similar to TestJet. Silkscreen markings have been added to the amp board to make it easier to get it right. In the above illustration, the sensor plate has been oriented such that the input pin is to the left and the ground pin is to the right. There is a trace from the input pin to the center via. The amp board shows the side that goes towards the sensor plate. Notice the "G" in the lower right corner. This matches the ground pin on the sensor plate. The "I" in the upper left corner matches the input pin on the sensor plate.





The above pictures show the two boards properly assembled. In these pictures, the connections to the mux are as follows. Ground is on the right. Signal is on the left.

Sizing Sensor Plates

TestJet guidelines for sensor plates instructed the fixture vendor to mill the plates nearly as big as the device outline. This gave the sensor plate maximum surface area for capacitive coupling. However, on today's boards, many BGAs are close to probes, bypass capacitors and other components that can erroneously couple to the sensor plate during the test. With VTEP's enhanced measurement capability, sensor plates can be made smaller to reduce this "fringing" effect, without sacrificing test quality.

Custom sensor plates can be milled from double-sided copper clad material to fit small geometry connectors and other devices. The diagram and photograph below show an L-shaped sensor plate designed to fit into a small connector and yet provide ample space to mount the amp board to the plate. When fabricating a custom sensor plate, connect the top side of the sensor plate to the ground pin of the amp probe. Connect the bottom side of the sensor plate to the signal pin of the amp probe. Use pins unsoldered from standard sensor plates to make these connections.

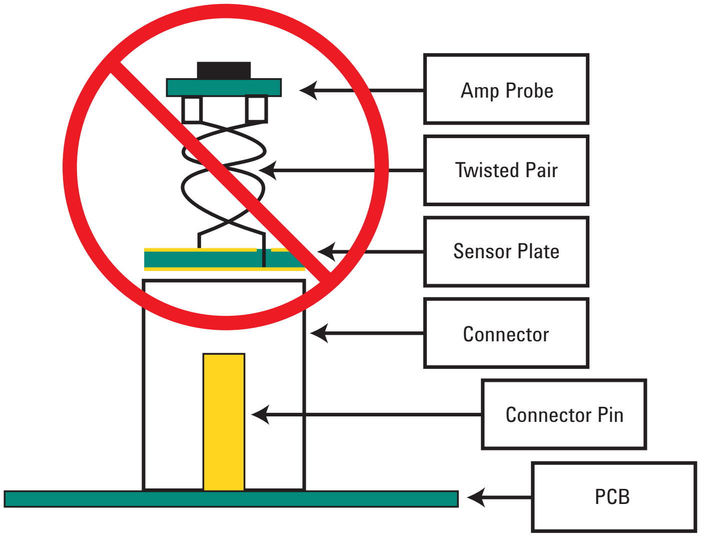
Provide a clearance hole around the signal pin in the top side of the sensor plate to be sure that the signal is not shorted to ground. Use solder to make sure that all similar planes are electrically connected.

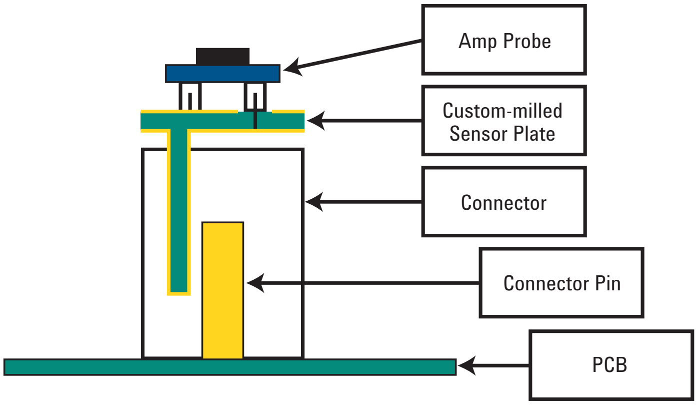
Finally, you can cut notches and drill holes in the sensor plates as needed, as long as you do not damage the amp boards.

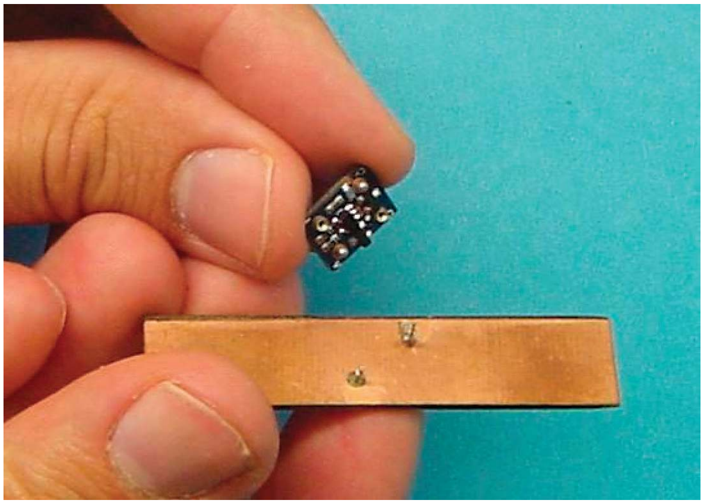
Customizing Sensor Plate

Warning: The amp board should always be pressed directly onto the sensor plate. Never insert wiring at this interface. Doing so can inject significant noise into the measurements. Proper mounting and wiring is extremely important when there are probes mounted on the same side of the fixture as the VTEP hardware. Often the probe wiring will couple to the wiring between the sensor plate and the amp and cause false readings.

Custom sensor plates can be milled from double-sided copper clad material to fit small geometry connectors and other devices. The diagram and photograph below show an L shaped sensor plate designed to fit into a small connector and yet provide ample space to mount the amp board to the plate.







Note that there is excellent capacitive coupling between the sensor plate and the connector pins. And more importantly, there is no twisted pair wiring to add measurement noise between the sensor plate and amp board.

When fabricating a custom sensor plate, connect the top side of the sensor plate to the ground pin of the amp probe. Connect the bottom side of the sensor plate to the signal pin of the amp probe. Use pins unsoldered from standard sensor plates to make these connections. Provide a clearance hole around the signal pin in the top side of the sensor plate to be sure that the signal is not shorted to ground. Use solder to make sure that all similar planes are electrically connected. Finally, you can cut notches and drill holes in the sensor plates as needed, as long as you don't damage the amp boards.

VTEP Part Numbers

Fixture Components	**TestJet P/N**	**VTEP P/N**
Mux	E3849A	N4300A
Mux + Ref	E3969A	N4307A
Probes (amp boards)	E3850A / E3960A	N4301A / N4313A
Connector Sensor Plates	E3964A	N4306A
1.2″ sq. Sensor Plates	E3853A	N4302A
2.5″ sq. Sensor Plates	E3854A	N4303A
Small probes (small amp boards)	E3845A / E3847A	N4311A / N4312A
B-C Polarity Check Sensor Plates	E3891A	
D Polarity Check Sensor Plates	E3892A	

Additional Fixturing Considerations for VTEP

VTEP uses different hardware than TestJet. While mechanically the same, there are important electrical differences that must be considered when fixturing. Failure to observe the following requirements could result in unstable VTEP and iVTEP measurements in production.

- Do not mix VTEP and TestJet muxes in the same fixture.
- Do not mix VTEP and TestJet probes in the same fixture.

VTEP requires that the mux to tester connections be maintained at all times. Also, minimizing the resistance in the VTEP sensor plate to mux wiring path is critical to the proper operation of VTEP.

Always:

- Solder the amp board to the hanger probe
- Make solder connections after verifying the proper assembly and operation of VTEP components.
- Use both the "verify all mux cards" AND "verify all testjet probes" commands to verify.

Never:

- Use transfer pins in the mux to sensor plate path.
- Use make-break connections (for example, crown probes) in the mux to sensor plate path.
- Use transfer pins in the mux to tester path.

At run-time, VTEP may give the following message if there is non-VTEP hardware or a potential fixturing problem:

> T406: Invalid Testjet probe, use "verify allmux cards" to debug

If the hardware is VTEP, then this message indicates a fixture problem. Check that the path connections are soldered, replace the hanger probes, or check the path from the mux header to the sensor plate for other sources of resistance.

Enhanced Guarding

Software release 7.20p includes an enhanced guarding feature. Enhanced guarding provides an improved guard path (lower impedance) for better VTEP and NPM measurements.

Enhanced guarding is specified by adding "throughput adjustment 4" to the Vtep source. The process to add enhanced guarding is:

1. Add "default throughput adjustment 4" to the Vtep source
2. Compile the modified test.
3. Run Auto Debug.

Why is it important?

Some DUTs have devices with pins that are connected through low-valued resistors to a fixed node. When these are guarded during the VTEP test, the small guard path impedance forms a voltage divider that allows the stimulus to leak into the fixed node. These many small signals sum together and cause the reading for the stimulus pin to be higher. This can mask an open.

What does it do?

A new option is available to close the hybrid card switched ground connections automatically when running a VTEP test. This dramatically reduces the guard path impedance and improves the VTEP reading. The new guarding will not impact throughput and will not cause a 'short to system ground' error.

When should it be used?

Enhanced guarding will improve all Vtep tests and should be used for every new development. Enhanced guarding should be added to existing boards when Auto Debug is run. After adding "throughput adjustment 4" to a test, some pins will have a lower nominal reading.

Will enhanced guarding be added automatically?

Enhanced guarding will automatically be added to all new tests but will not automatically be added to existing tests. New tests will have enhanced guarding added to the sources as the default during program generation. The user must run Auto Debug after adding enhanced guarding. Auto Debug will preserve but not add enhanced guarding.

What is the effect of enhanced guarding on iVtep?

Enhanced guarding is not used with iVtep and therefore has no effect.

AutoDebug

Quick Start

The AutoDebug tool (ADB) was designed to allow a test developer to quickly and easily get a VTEP test debugged and into production. ADB does not support TestJet. After the board is in production, a sustaining engineer can use the tool to quickly and easily adjust a test if needed. The tool is integrated into push-button debug and is invoked with a simple mouse click. The quick way to debug the test is:

- Place a known good board on the fixture
- Load the testplan into basic
- Execute debug *"testjet"*
- When the push button debug window appears, select the AutoDebug Test option from the *AutoDebug* menu.
- When ADB is finished the test is ready to run.

Several things happen while ADB is running: vacuum is activated, the test is run, measurement data is captured, the data is evaluated, a new TestJet test is written, the new test is compiled, and finally vacuum is released. The time for this is approximately 10 seconds plus 10 times the run time of the test (ten readings are taken for each pin during this process). ADB will have set limits for each pin and commented out any pins, which do not meet the acceptance criteria (minimum cpk, see below for explanation).

Assumptions & Use Model

ADB assumes you are debugging a Known Good Board. It tries to set limits such that this board will pass reliably in production. ADB takes 10 readings for each pin and calculates a mean and standard deviation. Limits are evaluated using two sets of rules. The more conservative (wider) of the two limits is chosen. See the Custom Parameters section for details on the rules.

After the first time ADB is run on a test, the old limits are taken into account when setting new limits. This allows debugging across multiple known good boards and fixtures. If a new limit is calculated outside the old limit, the new limit is used. If the old limit is outside the new limit, the old limit is kept. The old limit represents previous ADB runs on other boards. If the old limit is outside the newly calculated one this means that there is another board which needs the limit set wider and ADB should not bring it in closer.

Ideally, ADB should be run on several known good boards with devices from different manufacturer's lots and across each fixture if more than one is being built. This will take into account the fixture-to-fixture and board-to-board variations. Realistically, the programmer may have only one fixture and board available. In this case, run ADB again on the same board after opening and closing the fixture. This will help simulate any variations due to mechanical inconsistencies in sensor plate alignment in the fixture. After this, ADB can be run again after the board is in production. This may be the only way to cover all fixtures. Boards from a production run will also tend to have new lots of devices loaded onto the boards.

ADB has other options in the AutoDebug menu. These are summarized here along with intended/possible uses for them.

Menu Choice	Description
AutoDebug Test	As described above. Debugs the entire test.
AutoDebug Device	Debugs a single device selected from the device list. Use this if you want to debug only one device leaving the other devices unchanged.
Undo Last AutoDebug	Undo of last operation. Each time a change to the test is made, a numbered backup copy is made. The undo is implemented by overwriting the test with the last backup. This can be repeated multiple times. Each undo throws away the most recent version of the test, which means there is no redo. Backups are made in the <board dir>/debug/TestJet/adb directory.
Restore Original Test	The first time ADB is run, a golden copy of the test is saved. This will be the IPG generated version unless it has already been manually edited. This option first backs up the test and then restores the test to the golden copy. This is a way to start over, throw away any learned limits, and recover any pins that were commented out by ADB.
Restore Original Device	Same as above but for a single device selected from the device list. Use this after an ECO on the board that changes a device. You can keep all other learned limits and then debug just this new device.
Edit Custom Parameters	Some of the parameters used by ADB are customizable by the user. This will bring up a window for editing the parameters. After changing these, ADB can be run again using the changes. The parameters used by ADB are documented in the comments at the beginning of the test.
AutoDebug Test iVTEP + VTEP	As described above. Debugs the entire test with both VTEP and iVTEP.
AutoDebug Device iVTEP + VTEP	Debug a single device selected from the device list with both VTEP and iVTEP. Use this if you want to debug only one device leaving the other devices unchanged.

Custom Parameters

When determining limits for each pin, ADB uses two separate calculations and chooses the limit that is the most conservative (wider). These two calculations are done using an *internal* and a *user* formula.

The *internal* method uses empirically derived default characteristics for parameters such as noise and offsets due to temperature. The user has no control over this calculation. It provides a certain minimum margin between the mean and the limits that are set. This keeps ADB from writing limits that are closer than the measurement capability no matter what custom parameters are set.

The *user* formula is fully customizable by the user via several parameters. The values that the user can vary are accessed by the Edit Custom Parameters option in the Autodebug menu in Pushbutton Debug. There are two parameters for the high limit and the two for the low limit which function in a similar manner.

The **HiPercent** and **LoPercent** are multiplied times the mean to get an offset to add (or subtract) to the mean. These are not expressed as a true percent but as percent/100. In other words setting **LoPercent** to 0.50 would give an offset of 50% of the mean. As the mean gets lower, the offset from this calculation gets smaller. The **HiMinOffs**et and **LoMinOffset** values specify a minimum amount of offset from the mean to the limit. Expressed as a formula:

HiLimit = mean + max ((mean***HiPercent**), **HiMinOffset**)

LoLimit = mean – max ((mean***LoPercent**), **LoMinOffset**)

For the low limit another parameter controls the lowest limit set. No limit will be set lower than the **LowestLowLimit**.

The last parameter is **MinCpk**. This determines when pins are commented out of the test. After the limits are determined, the cpk is calculated. Any pin whose cpk falls below this threshold will be commented. *Note:* the cpk calculated uses the mean and standard deviation from this set of 10 measurements. It is not a cpk across all boards that have run through ADB. Cpk is used by many as a measurement of test quality in production and is defined by the following formula.

cpk = abs(mean – closest_limit) / (3 * sigma)

abs = absolute value

mean = average of the measurements

sigma = standard deviation

The formulas for calculating limits have been simulated in an Excel spreadsheet which is found in $KEYSIGHTICT_ROOT/contrib/doc/VectorlessEP_ADB.xls. The limits are calculated and graphed for low, medium, and high range values. This allows observation of the effects of the custom parameters.

AutoDebug Test iVTEP + VTEP

This is will be similar to the existing menu choices for AutoDebug Test but will include both VTEP and iVTEP.

This option makes a backup of the existing test, runs the test evaluates the results to calculate the new thresholds, and applies a CPK analysis. Pins that read below a minimum CPK then have the "throughput adjustment 2" added and their thresholds are set to 1 and 10. This turns on the iVTEP measurement for that pin. The syntax is as follows:

```
test pins "A3"; threshold low 2 high 28
!ADB:10,0.572,7.77,20.0,10000.0,3.36

test pins "A4"; threshold low 1 high 10,
throughput adjustment 2 !iVTEP
```

The test is compiled and run a second time and the data from only those pins with "throughput adjust 2" are analyzed. Those that are less than the minimum CPK are commented, the test is compiled for a final time. Summary data is shown in the workspace.

Faon/Faoff Macros

While ADB is running the board is pulled down and later released. This is done by executing the two macros *<board dir>/debug/TestJet/Faon & Faoff*. If your board is not purely vacuum actuated, these macros can be customized as needed. Example: you might do this if using an automated board handler or if the fixture actuation is pneumatic.

Comments

ADB writes comments into the test file when it runs. *DO NOT EDIT THIS INFORMATION*. It is important that these comments be left intact. While most of the comments are for information only, some of them are used to communicate to the next run of ADB. Modifying these comments can corrupt the information and cause incorrect behavior with ADB. Other comments can be inserted into the file safely by simply typing them on their own line.

Permanent Changes

ADB does not provide any mechanism to mark a device or pin as permanent. Each time it is run, ADB will rewrite all devices and pins. If you want to preserve a device (treat it as permanent), you must do this manually. Possible options are:

- Make a separate test for the permanent device(s). Don't run ADB on this test. Make sure to modify the testorder file as needed.
- Only use the AutoDebug Device option. This will only change the selected device and preserve all others.
- Keep a separate copy of the permanent device. Then you can manually restore it after ADB runs.

Remember IPG can overwrite the test unless it is marked permanent in the testorder file.

Other Macros

The following macros are in the Macros menu.

- TestJet Close Adjust-With Board
- TestJet Close Adjust-Without Board
- TestJet Close Adjust-Calculate Results

This set of macros measure and compare values with and without the board. The report generated gives recommendations for manual adjustments to some of the low thresholds to account for bottom side devices being close to probes in the fixture and measuring extra capacitance even if a pin is open.

The following macros are also in the Macros menu.

- TestJet Missing Adjust-Blank
- TestJet Missing Adjust-Loaded
- TestJet Missing Adjust-Calculate Results

This set of macros measure and compare the difference between a loaded and a blank board. The report generated gives recommendations for manual adjustments to some of the high thresholds to make sure the test fails if the device is missing. A sensor plate directly touching the bare board usually measures higher than normal readings when the device is missing.

The guidelines used by these macros were developed for TestJet. They have not been updated for VTEP. The recommendations they give are probably still worth considering but they have not been validated against the better performance characteristics of VTEP.

NOTE: Any manual changes made in the test file will be lost if ADB is run again. If these macros are to be used, run ADB first then make any manual changes. Keep the report files for future use if ADB needs to be run again.

Output Format

If you use the output of *testjet print level is al*l to collect data into the printer is device, there are some enhancements to the format of the printed data. The original format was:

```
D: u3 P: 3 25.3 p 13.0 46.0
```

This meant device (D) u3, pin (P) 3 measured 25.3fF with limits of 13 and 46. This is still valid. In the past, only pins with measurements were printed. Now tied pin information is also included as follows:

```
D: u3 P: 4 18.7 p 7.0 39.0
D: u3 P: 7 tied
```

This means pin 7 is tied to pin 4 of device u3.

If a pin is tested using a drive-through node the format changes to:

```
D: u3 N: NodeName 192.7 p 147.0
264.0
```

Notice the "N" for node instead of "P" for pin.

One last change, when using original TestJet a string of "-" characters are printed to separate devices. VTEP tests are separated with "=" characters.

Software Release 7.0 and VTEP v2.0

Software version 07.00p provides a common software platform for the both i3070 muxed and i3070 unmuxed test platforms. VTEP v2.0 test suite is also part of the i3070 SW version 07.00p. For VTEP-equipped fixtures, VTEP v2.0 extends VTEP with iVTEP for complete device signal pin coverage and NPM for connector power and ground coverage. Older TestJet fixturing is also supported.

NPM Connector Models

The version 07.00p software upgrade provides NPM for additional coverage on connector power and ground pins. NPM relies on the network parameter measurements of the connector to detect opens on power and ground pins. These network parameters are described in connector models. These connector models are supplied in the 3070 library structure beginning with the ver 07.00p software upgrade at:

$KEYSIGHTICT_ROOT/library/
supplemental/connectors

The following connectors have NPM models at ver 07.00p:

Description	Typical Manufacturer Part Numbers	3070 Connector Model Name
Fully Buffered DIMM (FBD), 240 pins DDR II	AT54007-H3BF-4F	fbd_ddr2_240_pin
DIMM, 240 pins DDR II	1775805-1	ddr2_240_pin
PCI Express 32X	2EG0E017-D2D-DF	pci_express_32x
PCI Express 16X	2EG38211-D2D-4F	pci_express_16x
PCI Express 8X	2EG08211-D2D-DF	pci_express_8x
PCI Express 4X	2EG03211-D2D-DF	pci_express_4x
PCI Express 1X	22EG01811-D2D-DF	pci_express_1x
Serial ATA 22 pins	LD11223-S03	sata_22_pin
Metral® 11Row	2H251103-NC201	metral_11_row
FH B-T-B 0.8mm, 80 pins	177983	08mmfh_80

The following CPU sockets have NPM models added from software version 07.10:

Description	Typical Manufacturer Part Numbers	3070 Connector Model Name
775 pin LGA CPU socket	PE077507-2041-01	lga_775
1207 pin LGA CPU socket	PE120703-3741	lga_1207

The connector models are encrypted. The pin names in the model are based on industry standard naming conventions for the connector. They must match the pin names in the board file. If not, a board compiler warning is generated. You can either edit the board file pin names or edit the map file (see description below) to fix this warning. At debug and runtime, a valid model file must be present. With the 7.20p release model files can be in any directory pointed to by the LIBRARY OPTIONS in the board file such as the board directory under CUSTOM_LIB or in the standard location on the testhead controller at:

$KEYSIGHTICT_ROOT/library/
supplemental/connectors

Additional Test Development Step for NPM

VTEP v2.0 uses the same test development process for TestJet, VTEP, and iVTEP plus an additional step NPM. This additional step is to add the connector model from the above table to part number field for the target connector. For i3070 muxed systems, you can use Board Consultant. For i3070 unmuxed systems, you can use TTFXT. In either case, the final entry in the board file will look like this:

CONNECTOR
“J601” TJ PN”pci_express_16x” “CONN-PC EDGE”; !VTEP+NPM connector

“J602” TJ “CONN-POST-TP-PST”; !VTEP-only connector

VTEP v2.0 File Structure

VTEP v2.0 uses the same file structure for TestJet, VTEP, and iVTEP plus additional files for NPM. These files are:

"testjet":

This is a testjet source file created by IPG. It includes all devices marked for vectorless testing. It also includes all connectors that are marked for vectorless testing and do NOT have an NPM connector model in the part number field in the board file.

For i3070 unmuxed systems with the "enable testjet" removed from the board config file, this file is named "vtep".

This file is later compiled for requirements and an object. There will be corresponding entries in testorder, testplan and wirelist.

It is debugged using ADB. After debugging with ADB, an ASCII file called "testjet.bd" is written with the per-pin mean values captured during that run of ADB. This called a "baseline data" file. It can be used to keep track of ADB pins. It is not required for TestJet, VTEP or iVTEP at runtime.

At runtime, this file is executed by the testplan. For TestJet-equipped fixtures it will make TestJet measurements. For VTEP-equipped fixtures, it will make VTEP and iVTEP measurements.

"testjet_gx":

This is a TestJet source file created by IPG. It includes all connectors that are marked for vectorless testing and have an NPM connector model in the part number field in the board file. It does NOT include devices and connectors found in the "testjet" file.

For i3070 unmuxed systems with the "enable testjet" removed from the board config file, this file is named **"vtep_gx"**.

This file is later compiled for requirements and an object. There will be corresponding entries in testorder, testplan and wirelist.

It is debugged using ADB. After debugging with ADB, an ASCII file called **"testjet_gx.bd"** is written with the per-pin mean values captured during that run of ADB. This is call a "baseline data" file. For NPM, this file must be present at runtime or an error will occur.

At runtime, this file is executed by the testplan. For TestJet-equipped fixtures it will make TestJet measurements. For VTEP-equipped fixtures, it will make VTEP and iVTEP measurements first. The test will terminate if there any VTEP or iVTEP failures. If these all pass, AND a valid mapfile, baseline data file and model file are present, NPM is executed to gain additional coverage of the power and ground pins.

"testjet_gx.m":

This is an ASCII file describing the model path and name, model pin name, the board pin name, the pintype and tied status for each connector in the "testjet_gx" file. It's called a "mapfile".

For i3070 unmuxed systems with the "enable testjet" removed from the board config file, this file is named **"vtep_gx.m"**.

You can edit the file if the connector pin names in the board file do not match the connector pins names in the model. These will be marked with a "*" in the mapfile.

You can edit the file and remove the individual connector power, ground, and fixed pins from NPM analysis by changing their pintype to "nc". No recompiling of the "testjet_gx" file is needed. However, you will need to execute a 'load board' command from the BT-Basictesthead window.

For NPM, this file must be present at runtime or an error will occur.

An example map file is shown below:

```
! Map file for “DEMO 2004 Board”
!-----------------------------------------------------------------
! Pin type can be one of:
!   gnd       ground
!   fixed     from board file fixed list
!   signal    signal pin
!   nc        not connected
!   na        no access
!
! Pin names that are defined in the board
! file but are not defined in the model
! file have the Model Pin Name defined with
! an “*” and require the user to fix this
! file to put in the correct Model Pin Name
! for the specified Board Pin Name.
!-----------------------------------------------------------------
device “j602”
model “C:/Keysight_ICT/library/supplemental/connectors/models/
pci_express_16x”

  !---------------------------------------------------------------
  ! Model   Board
  ! Pin     Pin       Pintype     Tied?
  ! Name    Name                  (1,2,4...)
  !---------------------------------------------------------------
    A1      A1        signal      A5, A6, A7, A8, A11, A15
    A2      A2        fixed
    A3      A3        fixed
    A4      A4        gnd
    A5      A5        signal      A1, A6, A7, A8, A11, A15
```

Test Development

The following table summarizes the VTEP v2.0 test development process. See the i3070 VTEP manual for additional details.

	***Medalist* i3070 muxed VTEP or TestJet**	***Medalist* i3070 unmuxed**	
		VTEP	**TestJet**
"config" file	*enable testjet* Enables TestJet, VTEP, iVTEP and NPM at 7.0	*!enable testjet* Enables VTEP, iVTEP and NPM at 7.0	*enable testjet* Enables TestJet only[1]
"board" file	Devices and non-NPM connectors – Normal vectorless test entry. NPM Connectors – Add connector model file name to the part number field for target connector.	Devices and non-NPM connectors – Normal vectorless test entry. NPM Connectors - Add connector model file name to the part number field for target connector.	Devices and non-NPM connectors – Normal vectorless test entry. Future NPM Connectors – Add connector model file name to the part number field for target connector.
Test Generation	Creates the following files: "testjet": All devices and connectors without connector model specified "testjet_gx": Connectors with connector model specified. "testjet_gx.m": Connector map file.	Creates the following files: "vtep": All devices and connectors without connector model specified. "vtep_gx": Connectors with connector model specified. "vtep_gx.m": Connector map file	Creates the following files: "testjet": All devices and connectors without connector model specified "testjet_gx": Connectors with connector model specified. "testjet_gx.m": Connector map file
"testorder", "testplan", "wirelist" files	The following test entries are added: "testjet": All devices and connectors without connector model specified. "testjet_gx": Connectors with connector model specified.	The following test entries are added: "vtep": All devices and connectors without connector model specified. "vtep_gx": Connectors with connector model specified.	The following test entries are added: "testjet": All devices and connectors without connector model specified. "testjet_gx": Connectors with connector model specified.

1. NPM and iVTEP require VTEP hardware. TestJet hardware does not support NPM. However, test fixtures with TestJet can be easily upgraded with VTEP in the future. For these cases, you can specify NPM connector models so that the file structure of the board directory will be in place when you upgrade the fixture hardware. To enable iVTEP capability after such an upgrade, restore the original test and autodebug.

Debug

The following table summarizes the VTEP v2.0 debug process. See the i3070 VTEP manual for additional details.

	i3070 muxed VTEP or TestJet	i3070 unmuxed	
		VTEP	TestJet
ADB – VTEP HW	*debug “testjet”:* AutoDebug Device AutoDebug Test AutoDebug VTEP+iVTEP Device AutoDebug VTEP+iVTEP Test “testjet.bd file” created *debug “testjet_gx”:* AutoDebug Device AutoDebug Test “testjet_gx.bd” file created	*debug “vtep”:* AutoDebug Device AutoDebug Test AutoDebug VTEP+iVTEP Device AutoDebug VTEP+iVTEP Test “vtep.bd” file created *debug “vtep_gx”:* AutoDebug Device AutoDebug Test “vtep_gx.bd” file created	*debug “testjet”:* Config target statement does not match hardware. Error generated. *debug “testjet_gx”:* Config target statement does not match hardware. Error generated
ADB – TestJet HW	*debug “testjet”:* ADB is not available for TestJet on the i3070 muxed system. Warning generated. *debug “testjet_gx”:* ADB is not available for TestJet on the i3070 muxed system. Warning generated.	*debug “vtep”:* Config target statement does not match hardware. Error generated. *debug “vtep_gx”:* Config target statement does not match hardware. Error generated	*debug “testjet”:* AutoDebug Device AutoDebug Test “testjet.bd” file created *debug “testjet_gx”:*[1] AutoDebug Device AutoDebug Test “testjet.bd” file created A warning will be generated alerting the user to the better performance available if the fixture was equipped with VTEP.

[1] The testjet_gx file will only cover signal pins for TestJet-equipped fixtures. NPM measurements for opens coverage of connector power and ground pins will not be made.

Runtime

The following table summarizes VTEP v2.0 runtime execution. See the i3070 VTEP manual for additional details.

	i3070 muxed VTEP or TestJet	i3070 unmuxed	
		VTEP	TestJet
Runtime – VTEP HW	*test "testjet"*: Test device signal pins with VTEP or iVTEP Test non-NPM connector signal pins with VTEP *test "testjet_gx"*: Test NPM connector signal pins with VTEP. The test terminates prior to NPM if any signal pins fail. Execute NPM on power and ground pins if all VTEP signal pins pass. The connector model file, map file and baseline data file must be available or a runtime error will occur.	*test "vtep"*: Test device signal pins with VTEP or iVTEP Test non-NPM connector signal pins with VTEP *test "vtep_gx"*: Test NPM connector signal pins with VTEP. The test terminates prior to NPM if any signal pins fail. Execute NPM on power and ground pins if all VTEP signal pins pass. The connector model file, map file and baseline data file must be available or a runtime error will occur.	*test "testjet"*: Config target statement does not match hardware. Error generated. *test "testjet_gx"*: Config target statement does not match hardware. Error generated.
Runtime – TestJet HW	*test "testjet"*: Test device signal pins with TestJet Test non-NPM connector signal pins with TestJet *test "testjet_gx"*: Test NPM connector signal pins with TestJet.	*test "vtep"*: Config target statement does not match hardware. Error generated. *test "vtep_gx"*: Config target statement does not match hardware. Error generated.	*test "testjet"*: Test device signal pins with TestJet Test non-NPM connector signal pins with TestJet *test "testjet_gx"*: Test NPM connector signal pins with TestJet.

Additional NPM Debug Considerations

Always use ADB to debug the "testjet_gx" file on a known good board (KGB). Remember that VTEP measurements are made for signal pins first. If these pass, then NPM is executed for the power, ground and fixed pins as defined in the connector model and mapfile.

To remove individual power, grounds, or fixed pins from NPM execution, change the mapfile entry for that pin to "nc". In the example below, pin "A2" will be removed from NPM execution:

```
device "j602"
model"C:/Keysight_ICT/library/supplemental/connectors/models/
pci_express_16x"

 !------------------------------------------------------------------------------------
 ! Model    Board
 ! Pin      Pin       Pintype    Tied?
 ! Name     Name                 (1,2,4...)
 !------------------------------------------------------------------------------------
   A1       A1        signal     A5, A6, A7, A8, A11, A15
 ! A2       A2        fixed
   A2       A2        nc
```

No recompiling of the "testjet_gx" file is needed. However, you will need to execute a 'load board' command from the BT-Basictesthead window.
To remove NPM execution for an entire connector, remove the model name entry from the part number filed in the board file, re-compile board and re-compile "testjet_gx".

Miscellany

CR14114

In some cases where a pin is tied to another through a jumper, the output of the *testjet print level* is all feature is incorrect. When this happens, ADB comments out the pin. To workaround this, manually uncomment the pin and fix the tied pin syntax by looking at the original test. Look for the comment *"!ADB CR"* at the beginning of the line to identify these pins.

CR13647

The TestJet test compiler can now compile tests that formerly were too large to compile. However, a new problem has appeared. Tests that are too large (this is larger than the previous limit) can cause *a T313 timeout waiting on psync* exception. The workaround to this is to execute "dump flow control off" just prior to executing a vtep test when nrun = 1 followed by "dump flow control off" just after executing the test. Do nothing (leave off) on nrun > 1.

myKeysight

www.keysight.com/find/mykeysight
A personalized view into the information most relevant to you.

www.axiestandard.org
AdvancedTCA® Extensions for Instrumentation and Test (AXIe) is an open standard that extends the AdvancedTCA for general purpose and semiconductor test. Keysight is a founding member of the AXIe consortium. ATCA®, AdvancedTCA®, and the ATCA logo are registered US trademarks of the PCI Industrial Computer Manufacturers Group.

www.lxistandard.org
LAN eXtensions for Instruments puts the power of Ethernet and the Web inside your test systems. Keysight is a founding member of the LXI consortium.

www.pxisa.org

PCI eXtensions for Instrumentation (PXI) modular instrumentation delivers a rugged, PC-based high-performance measurement and automation system.

Three-Year Warranty
www.keysight.com/find/ThreeYearWarranty
Keysight's commitment to superior product quality and lower total cost of ownership. The only test and measurement company with three-year warranty standard on all instruments, worldwide.

Keysight Assurance Plans
www.keysight.com/find/AssurancePlans
Up to five years of protection and no budgetary surprises to ensure your instruments are operating to specification so you can rely on accurate measurements.

www.keysight.com/quality
Keysight Technologies, Inc.
DEKRA Certified ISO 9001:2008
Quality Management System

Keysight Channel Partners
www.keysight.com/find/channelpartners
Get the best of both worlds: Keysight's measurement expertise and product breadth, combined with channel partner convenience.

www.keysight.com/see/vtep

For more information on Keysight Technologies' products, applications or services, please contact your local Keysight office. The complete list is available at:
www.keysight.com/find/contactus

Americas	
Canada	(877) 894 4414
Brazil	55 11 3351 7010
Mexico	001 800 254 2440
United States	(800) 829 4444
Asia Pacific	
Australia	1 800 629 485
China	800 810 0189
Hong Kong	800 938 693
India	1 800 112 929
Japan	0120 (421) 345
Korea	080 769 0800
Malaysia	1 800 888 848
Singapore	1 800 375 8100
Taiwan	0800 047 866
Other AP Countries	(65) 6375 8100
Europe & Middle East	
Austria	0800 001122
Belgium	0800 58580
Finland	0800 523252
France	0805 980333
Germany	0800 6270999
Ireland	1800 832700
Israel	1 809 343051
Italy	800 599100
Luxembourg	+32 800 58580
Netherlands	0800 0233200
Russia	8800 5009286
Spain	0800 000154
Sweden	0200 882255
Switzerland	0800 805353 Opt. 1 (DE) Opt. 2 (FR) Opt. 3 (IT)
United Kingdom	0800 0260637

For other unlisted countries:
www.keysight.com/find/contactus
(BP-07-10-14)

This information is subject to change without notice.

Published in USA, July 31, 2014
5989-6344EN
www.keysight.com